

WEST Search History

DATE: Saturday, June 10, 2006

| Hide? | <u>Set</u> <u>Name</u> | <u>Query</u> | <u>Hit</u> <u>Count</u> |
|--------------------------|-------------------------------------|--|----------------------------|
| | <i>DB=PGPB,USPT; PLUR=NO; OP=OR</i> | | |
| <input type="checkbox"/> | L30 | L28 and (messag\$ with version\$ with copy\$ with cach\$) | 4 |
| <input type="checkbox"/> | L29 | L28 and (messag\$ near version\$ near copy\$) | 0 |
| <input type="checkbox"/> | L28 | 707/200-203.ccls. | 5224 |
| <input type="checkbox"/> | L27 | 707/104.1.ccls. | 5128 |
| <input type="checkbox"/> | L26 | 707/8.ccls. | 1033 |
| <input type="checkbox"/> | L25 | (cach\$ with (recent near copy) with message\$) | 6 |
| <input type="checkbox"/> | L24 | L23 and messag\$ | 16 |
| <input type="checkbox"/> | L23 | L21 and copy\$ | 19 |
| <input type="checkbox"/> | L22 | L21 and (most near recent near cop\$) | 0 |
| <input type="checkbox"/> | L21 | (L19 or L20) and (local near cach\$) | 20 |
| <input type="checkbox"/> | L20 | ((multi adj1 system\$) or (multiple adj1 system\$) or multiple-system\$ or multi-system\$.ab. | 316 |
| <input type="checkbox"/> | L19 | ((multi adj1 system\$) or (multiple adj1 system\$) or multiple-system\$ or multi-system\$.ti. | 146 |
| | <i>DB=USPT; PLUR=NO; OP=OR</i> | | |
| <input type="checkbox"/> | L18 | L17 and cach\$ | 8 |
| <input type="checkbox"/> | L17 | L16 and version\$ | 16 |
| <input type="checkbox"/> | L16 | (L14 or L15) and messag\$ | 49 |
| <input type="checkbox"/> | L15 | (multiple adj1 system\$.ab. | 138 |
| <input type="checkbox"/> | L14 | (multiple adj1 system\$.ti. | 46 |
| <input type="checkbox"/> | L13 | L12 and cach\$ | 5 |
| <input type="checkbox"/> | L12 | L11 and messag\$ | 6 |
| <input type="checkbox"/> | L11 | L10 and (multiple adj1 system\$) | 10 |
| <input type="checkbox"/> | L10 | L4 and version\$ | 68 |
| <input type="checkbox"/> | L9 | L8 and cach\$ | 4 |
| <input type="checkbox"/> | L8 | L7 and version\$ | 4 |
| <input type="checkbox"/> | L7 | L6 and messag\$ | 4 |
| <input type="checkbox"/> | L6 | L4 and (local adj1 cach\$) | 4 |
| <input type="checkbox"/> | L5 | L4 and ((messag\$ or email\$ or e-mail\$ or (electronic adj1 mail\$) or (text adj1 messag\$) or text-messag\$) with version\$ with cach\$) | 1 |

(5543236 5931938 4800481 4993017 5280611 5327402 5333188 5448720)

10/12/2005

| | | |
|--------------------------|--|-----|
| | 5469468 5512538 5583093 5724648 5735215 5784274 5889504 6025775 6084871 6131166 6237020 4257100 4271508 4301379 4513224 4557747 4571529 4593499 4954715 4996718 5248773 5282020 5327576 5390329 5403681 5412091 5428470 5505172 5507288 5534845 5545498 5551046 5555369 5570372 5591023 5646981 5689810 5691630 5712989 5713395 5732192 5753054).pn. (5760936 5805870 5828573 5841963 5861283 5870306 5894341 5896510 5900806 5930831 5930361 5963731 5970408 5978353 6006106 6086651 6105016 6116912 6126310 6173347 6195559 6219288 6219288 6227985 6237041 6249150 6249893 4300603 4304001 4306288 4322801 4331834 4333884 4340775 4340776 4352103 4376127 4397289 4402082 4410225 4412281 4465416 4470681 4477599 4513699 4533470 4532904 4535451 4536864 4569048).pn. (4568963 4571639 4584639 4603968 4607913 4611197 4612637 4612542 4622664 4628529 4630305 4630304 4758794 4766501 4779937 4799253 4809059 4809267 4812447 4833701 4835685 4835903 4849489 4851995 4870610 4887297 4888913 4893327 4903321 4905181 4905507 4914583 4922480 4928880 4929466 4942406 4972362 4975865 4974444 4982352 5001625 5003468 5015054 5023909 5185879 5209653 5218696 5230019 5230794 5235633).pn. (5235854 5249206 5250041 5255537 5258714 5265250 5267883 5268240 5271453 5273024 5274838 5280424 5293871 5317568 5327556 5339330 5339427 5339450 5342329 5352657 5357561 5357614 5359417 5361199 5372408 5392403 5396388 5396552 5401713 5404501 5408017 5408653 5416921 5418637 5420917 5423664 5423768 5428666 5431016 5432421 5434691 5440918 5442785 5446553 5450551 5458404 5464038 5465362 5467940 5471460).pn. (5471674 5473596 5473608 5473682 5477128 5479736 5485425 5493508 5495239 5495343 5499373 5504557 5506787 5508912 5513005 5515169 5517622 5513312 5535334 5537457 5544334 5546445 5548525 5547675 5551073 5552764 5557792 5561798 5572529 5575634 5581701 5581750 5581753 5585332 5586338 5594890 5598863 5603029 5610943 5609747 5613068 5613204 5615235 5615290 5616743 5619641 5623666 5630163 5634193 5646521).pn. (5655130 5657128 5657440 5673039 5675628 5680609 5684701 5692142 5694470 5694401 5700999 5701909 5711299 5713353 5726981 5737703 5737726 5740092 5740424 5744787 5748124 5751575 5754104 5761618 5761660 5765198 5768375 5771185 5771225 5779226 5781614 5784278 5783989 5787122 5787114 5790073 5802149 5805334 5812622 5815814 RE35916 5822205 5825435 5828483 5827946 5829033 5835738 5838315 5850562 5852610).pn. | 299 |
| <input type="checkbox"/> | L3 L1 and document\$ | 0 |
| <input type="checkbox"/> | L2 L1 and (time or timestamp or date\$ or month\$ or year\$) | 1 |
| <input type="checkbox"/> | L1 5551046.pn. | 1 |

END OF SEARCH HISTORY

Dialog DataStar



options

logout

feedback

help

databases

easy search

Advanced Search:

Inspec - 1898 to date (INZZ)

limit

Search history:

| No. | Database | Search term | Info added since | Results | |
|-----|----------|---|------------------|---------|-----------------------------|
| 1 | INZZ | cache WITH message WITH version WITH copy | unrestricted | 0 | - |
| 2 | INZZ | cache WITH copy WITH message | unrestricted | 2 | show titles |
| 3 | INZZ | cache WITH copy | unrestricted | 99 | show titles |
| 4 | INZZ | 3 AND message | unrestricted | 5 | show titles |

[hide](#) | [delete all search steps...](#) | [delete individual search steps...](#)
Enter your search term(s): [Search tips](#) ☐ Thesaurus mapping
 whole document

 Information added since: or: none
 (YYYYMMDD)

search

☐ Documents with images

Select special search terms from the following list(s):

- ☒ Publication year 1950-
- ☒ Publication year 1898-1949
- ☒ Inspec thesaurus - browse headings A-G
- ☒ Inspec thesaurus - browse headings H-Q
- ☒ Inspec thesaurus - browse headings R-Z
- ☒ Inspec thesaurus - enter a term
- ☒ Classification codes A: Physics, 0-1
- ☒ Classification codes A: Physics, 2-3
- ☒ Classification codes A: Physics, 4-5
- ☒ Classification codes A: Physics, 6
- Classification codes A: Physics, 7

10/126,295

Dialog DataStar

options

logout

feedback

help

databases

search
page

titles

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document

search strategy: do not include the search strategy

order

☒ Select All

1 Architectural extensions to support efficient communication using message

2 Pin-down cache: a virtual memory management technique for zero-copy communi

☒ document 1 of 2 [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0007388169 20051201.

Title

Architectural extensions to support efficient communication using **message** prediction.

Conference information

Proceedings 16th Annual International Symposium on High Performance Computing Systems and Applications, Moncton, NB, Canada, 16-19 June 2002.

Source

Proceedings 16th Annual International Symposium on High Performance Computing Systems and Applications, 2002, p. 20-7, 25 refs, pp. xi +314, ISBN: 0-7695-1626-2.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

Author(s)

Afsahi-A, Dimopoulos-N-J.

Editor(s): Almhana-J-N, Bhavsar-V.





Author affiliation

Afsahi, A., Dept. of Electr. & Comput. Eng., Queen's Univ., Kingston, Ont., Canada.

Abstract

With increasing uniprocessor and SMP computation power workstation clusters are becoming viable alternatives to high performance computing systems. Communication overhead affects the performance of parallel computers significantly. A significant portion of the software communication overhead is attributable to **message** copying. We argue that it is possible to address the **message** copying problem at the receiving side through speculation. We show that messages display a form of locality, and we introduce the notion of **message** prediction for the receiving side of **message-passing** systems. By predicting a receive communication call before it is posted, we are able to place the required **message** directly into the **cache** speculatively before it is needed so that effectively a **zero-copy** communication can be achieved. Specific extensions to the ISA and the processor architecture accommodate late binding without requiring copying of the **message**.

Descriptors

 MESSAGE-PASSING;  PARALLEL-ARCHITECTURES;  PARALLEL-MACHINES;  PARALLEL-PROCESSING.

Classification codes

C5220P Parallel-architecture*;

C6150N Distributed-systems-software;
C5440 Multiprocessing-systems.

Keywords

SMP; workstation-clusters; high-performance-computing-systems; **message-copying**; parallel-computers; **message-prediction**; **message-** passing; **zero-copy-communication**.

Treatment codes

P. Practical.

Language

English.

Publication type

Conference-proceedings.

Availability

CCCC: 0-7695-1626-2/02/\$17.00.

Publication year

2002.

Publication date

20020000.

Edition

2002037.

Copyright statement

Copyright 2002 IEE.

COPYRIGHT BY The IET, Stevenage, UK

☒ **document 2 of 2** Order Document

Inspec - 1898 to date (INZZ)**Accession number & update**

0005907528 20051201.

Title

Pin-down **cache**: a virtual memory management technique for **zero-copy** communication.

Conference information

Proceedings of the First Merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing, Orlando, FL, USA, 30 March-3 April 1998.

Sponsor(s): IEEE Comput. Soc. Tech. Committee on Parallel Process; ACM SIGARCH.

Source

Proceedings of the First Merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing (Cat. No.98TB100227), 1998, p. 308-14, 9 refs, pp. xxv+809, ISBN: 0-8186-8404-6.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

Author(s)

Tezuka-H, O-Carroll-F, Hori-A, Ishikawa-Y.

Abstract

The overhead of copying data through the central processor by a **message** passing protocol limits data transfer bandwidth. If the network interface directly transfers the user's memory to the network by issuing DMA, such data copies may be eliminated. Since the DMA facility accesses the physical memory address space, user virtual memory must be pinned down to a physical memory location before the **message** is sent or received. If each **message** transfer involves pin- down and release kernel primitives, **message** transfer bandwidth will decrease since those primitives are quite expensive. The authors propose a zero **copy message** transfer with a pin-down **cache** technique which reuses the pinned-down area to decrease the number of calls to pin-down and release primitives. The proposed facility has been implemented in the PM low-level communication library on the RWC PC Cluster II, consisting of 64 Pentium Pro 200 MHz CPUs connected by a Myricom Myrinet network, and running NetBSD. The PM achieves 108.8 MBytes/sec for a 100% pin-down **cache** hit ratio and 78.7 MBytes/sec for all pin-down **cache** miss. The MPI library has been implemented on top of PM. According to the NAS parallel benchmarks result, an application is still better performance in case that **cache** miss ratio is very high.

Descriptors

 CACHE-STORAGE;  MESSAGE-PASSING;  PARALLEL-PROCESSING;  VIRTUAL-STORAGE.

Classification codes

C6120 File-organisation*;
C5320G Semiconductor-storage;
C6150N Distributed-systems-software.

Keywords

pin-down-cache; virtual-memory-management-technique; **zero-copy**- communication; data-copying-overhead; central-processor; **message**- passing-protocol; data-transfer-bandwidth; network-interface; user-memory-transfer; network; physical-memory-address-space; user-virtual-memory; physical-memory-location; release-kernel-primitives; pin-down-primitives; **message-transfer-bandwidth**; **zero-copy-message-transfer**; pinned-down-area-reuse; PM-low-level-communication-library; RWC-PC-Cluster-II; Pentium-Pro-CPU's; Myricom-Myrinet-network; NetBSD; pin- **down-cache-hit-ratio**; **pin-down-cache-miss**; MPI-library; NAS-parallel- benchmarks; 78.7-MByte/s; 108.8-MByte/s; 200-MHz.

Treatment codes

P Practical.

Numerical indexing

byte rate: 7.87E07 Byte/s.
byte rate: 1.088E08 Byte/s.
frequency: 2.0E08 Hz.

Language

English.

Publication type

Conference-proceedings.

Availability

CCCC: 1063-7133/98/\$10.00.

Digital object identifier

10.1109/IPPS.1998.669932.

Publication year

1998.

Publication date

19980000.



Edition

1998018.

Copyright statement

Copyright 1998 IEE.

COPYRIGHT BY The IET, Stevenage, UK

| | | |
|---|---|--|
|  | locally as: <input type="text" value="PDF document"/> | search strategy: <input type="text" value="do not include the search strategy"/> |
|  | | |

Top - News & FAQs - Dialog

© 2006 Dialog

Dialog DataStar

options

logout

feedback

help

databases

search
page

titles

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document

search strategy: do not include the search strategy

order

☒ Select All

- 1 [The use of prediction for accelerating upgrade misses in cc-NUMA multiproc](#)
- 2 [Architectural extensions to support efficient communication using message](#)
- 3 [Pin-down cache: a virtual memory management technique for zero-copy communi](#)
- 4 [Dynamic self-invalidation: reducing coherence overhead in shared- memory mu](#)
- 5 [The Kyushu University reconfigurable parallel processor-cache architec](#)

☒ document 1 of 5 [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0007503575 20051201.

Title

The use of prediction for accelerating upgrade misses in cc-NUMA multiprocessors.

Conference information

Proceedings 2002 International Conference on Parallel Architectures and Compilation Techniques. PACT 2002, Charlottesville, VA, USA, 22-25 Sept. 2002.

Sponsor(s): IEEE Comput. Soc. Tech. Committee on Comput Archit; IEEE Comput. Soc. Tech. Committee on Parallel Process; ACM Special Interest Group on Comput. Architecture; IFIP WG 10.3; IBM; Microsoft Res; Univ. Virginia Dept. Comput. Sci.

Source

Proceedings 2002 International Conference on Parallel Architectures and Compilation Techniques. PACT 2002, 2002, p. 155-64, 28 refs, pp. xviii+305, ISBN: 0-7695-1620-3.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

Author(s)

[Acacio-M-E](#), [Gonzalez-J](#), [Garcia-J-M](#), [Duato-J](#).

Author affiliation

Acacio, M.E., Gonzalez, J., Garcia, J.M., Duato, J., Univ. de Murcia, Spain.

Abstract

This work is focused on accelerating upgrade misses in cc-NUMA multiprocessors. These misses are caused by store instructions for which a read-only **copy** of the line is found in the L2 **cache**. Upgrade misses require a **message** sent from the missing node to the directory, a directory lookup in order to find the set of sharers, invalidation messages being sent to the sharers and responses to the invalidations being sent back. Therefore, the penalty paid by these misses is not negligible, mainly if we consider that they account for a high percentage of the total miss rate. We propose the use of prediction as a means of providing cc-NUMA multiprocessors with a more efficient support for upgrade misses by directly invalidating sharers from the missing node. Our proposal comprises an effective prediction scheme achieving high hit rates as well as a coherence protocol extended to support the use of prediction. Our work is motivated by two key observations: first, upgrade misses present a repetitive behavior and, second, the total number of sharers being invalidated is small (one, in some cases).

Using execution-driven simulations, we show that the use of prediction can significantly accelerate upgrade misses (latency reductions of more than 40% in some cases). These important improvements translate into speed-ups on application performance up to 14%. Finally, these results can be obtained including a predictor with a total size of less than 48 KB in every node.

Descriptors

 CACHE-STORAGE;  DELAYS;  MEMORY-PROTOCOLS;  SHARED-MEMORY-SYSTEMS.

Classification codes

C5640 Protocols*;
C5220P Parallel-architecture;
C6120 File-organisation.

Keywords

prediction; upgrade-miss-acceleration; cc-NUMA-multiprocessors; L2- **cache**; direct-invalidation; coherence-protocol; repetitive-behavior; sharers; execution-driven-simulations; latency-reductions.

Treatment codes

P. Practical.

Language

English.

Publication type

Conference-proceedings.

Availability

CCCC: 1089-795X/02/\$17.00.

Digital object identifier

10.1109/PACT.2002.1106014.

Publication year

2002.

Publication date

20020000.

Edition

2003002.

Copyright statement

Copyright 2003 IEE.

COPYRIGHT BY The IET, Stevenage, UK

 **document 2 of 5** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0007388169 20051201.

Title

Architectural extensions to support efficient communication using **message** prediction.

Conference information

Proceedings 16th Annual International Symposium on High Performance Computing Systems and Applications, Moncton, NB, Canada, 16-19 June 2002.

Source

Proceedings 16th Annual International Symposium on High Performance Computing Systems and Applications, 2002, p. 20-7, 25 refs, pp. xi +314, ISBN: 0-7695-1626-2.
Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

Author(s)

Afsahi-A, Dimopoulos-N-J.
Editor(s): Almhana-J-N, Bhavsar-V.

Author affiliation

Afsahi, A., Dept. of Electr. & Comput. Eng., Queen's Univ., Kingston, Ont., Canada.

Abstract

With increasing uniprocessor and SMP computation power workstation clusters are becoming viable alternatives to high performance computing systems. Communication overhead affects the

performance of parallel computers significantly. A significant portion of the software communication overhead is attributable to **message** copying. We argue that it is possible to address the **message** copying problem at the receiving side through speculation. We show that messages display a form of locality, and we introduce the notion of **message** prediction for the receiving side of **message-passing** systems. By predicting a receive communication call before it is posted, we are able to place the required **message** directly into the **cache** speculatively before it is needed so that effectively a **zero-copy** communication can be achieved. Specific extensions to the ISA and the processor architecture accommodate late binding without requiring copying of the **message**.

Descriptors

MESSAGE-PASSING; PARALLEL-ARCHITECTURES; PARALLEL-MACHINES; PARALLEL-PROCESSING.

Classification codes

C5220P Parallel-architecture*;
C6150N Distributed-systems-software;
C5440 Multiprocessing-systems.

Keywords

SMP; workstation-clusters; high-performance-computing-systems; **message-copying**; parallel-computers; **message-prediction**; **message-passing**; **zero-copy-communication**.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-proceedings.

Availability

CCCC: 0-7695-1626-2/02/\$17.00.

Publication year

2002.

Publication date

20020000.

Edition

2002037.

Copyright statement

Copyright 2002 IEE.

COPYRIGHT BY The IET, Stevenage, UK

☒ document 3 of 5 Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0005907528 20051201.

Title

Pin-down **cache**: a virtual memory management technique for **zero-copy** communication.

Conference information

Proceedings of the First Merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing, Orlando, FL, USA, 30 March-3 April 1998.
Sponsor(s): IEEE Comput. Soc. Tech. Committee on Parallel Process; ACM SIGARCH.

Source

Proceedings of the First Merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing (Cat. No.98TB100227), 1998, p. 308-14, 9 refs, pp. xxv+809, ISBN: 0-8186-8404-6.
Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

Author(s)

Tezuka-H, O-Carroll-F, Hori-A, Ishikawa-Y.

Abstract

The overhead of copying data through the central processor by a **message** passing protocol limits data transfer bandwidth. If the network interface directly transfers the user's memory to the network by issuing DMA, such data copies may be eliminated. Since the DMA facility accesses the physical memory address space, user virtual memory must be pinned down to a physical memory location before the **message** is sent or received. If each **message** transfer involves pin-down and release kernel primitives, **message** transfer bandwidth will decrease since those primitives are quite expensive. The authors propose a zero **copy message** transfer with a pin-down **cache** technique which reuses the pinned-down area to decrease the number of calls to pin-down and release primitives. The proposed facility has been implemented in the PM low-level communication library on the RWC PC Cluster II, consisting of 64 Pentium Pro 200 MHz CPUs connected by a Myricom Myrinet network, and running NetBSD. The PM achieves 108.8 MBytes/sec for a 100% pin-down **cache** hit ratio and 78.7 MBytes/sec for all pin-down **cache** miss. The MPI library has been implemented on top of PM. According to the NAS parallel benchmarks result, an application is still better performance in case that **cache** miss ratio is very high.

Descriptors

 CACHE-STORAGE;  MESSAGE-PASSING;  PARALLEL-PROCESSING;  VIRTUAL-STORAGE.

Classification codes

C6120 File-organisation*;
C5320G Semiconductor-storage;
C6150N Distributed-systems-software.

Keywords

pin-down-cache; virtual-memory-management-technique; **zero-copy**- communication; data-copying-overhead; central-processor; **message**- passing-protocol; data-transfer-bandwidth; network-interface; user-memory-transfer; network; physical-memory-address-space; user-virtual-memory; physical-memory-location; release-kernel-primitives; pin-down-primitives; **message-transfer-bandwidth**; **zero-copy-message-transfer**; pinned-down-area-reuse; PM-low-level-communication-library; RWC-PC-Cluster-II; Pentium-Pro-CPU's; Myricom-Myrinet-network; NetBSD; pin- **down-cache-hit-ratio**; **pin-down-cache-miss**; MPI-library; NAS-parallel- benchmarks; 78.7-MByte/s; 108.8-MByte/s; 200-MHz.

Treatment codes

P Practical.

Numerical indexing

byte rate: 7.87E07 Byte/s.
 byte rate: 1.088E08 Byte/s.
 frequency: 2.0E08 Hz.

Language

English.

Publication type

Conference-proceedings.

Availability

CCCC: 1063-7133/98/\$10.00.

Digital object identifier

10.1109/IPPS.1998.669932.

Publication year

1998.

Publication date

19980000.

Edition

1998018.

Copyright statement

Copyright 1998 IEE.

COPYRIGHT BY The IET, Stevenage, UK

☒ **document 4 of 5** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0005086781 20051201.

Title

Dynamic self-invalidation: reducing coherence overhead in shared-memory multiprocessors.

Conference information

Proceedings 22nd Annual International Symposium on Computer Architecture, Santa Margherita Ligure, Italy, 22-24 June 1995.

Sponsor(s): ACM SIGARCH; IEEE Comput. Soc., TCCA; Univ. Genoa.

Source

Proceedings 22nd Annual International Symposium on Computer Architecture (IEEE Cat. No.95CB35801), 1995, p. 48-59, 40 refs, pp. xiii+426, ISBN: 0-89791-698-0.

Publisher: ACM, New York, NY, USA.

Author(s)Lebeck-A-R, Wood-D-A.**Author affiliation**

Lebeck, A.R., Wood, D.A., Dept. of Comput. Sci., Wisconsin Univ., Madison, WI, USA.

Abstract

The paper introduces dynamic self-invalidation (DSI), a new technique for reducing **cache** coherence overhead in shared-memory multiprocessors. DSI eliminates invalidation messages by having a processor automatically invalidate its local **copy** of a **cache** block before a conflicting access by another processor. Eliminating invalidation overhead is particularly important under sequential consistency: where the latency of invalidating outstanding copies can increase a program's critical path. DSI is applicable to software, hardware, and hybrid coherence schemes. We evaluate DSI in the context of hardware directory-based write-invalidate coherence protocols. Our results show that DSI reduces execution time of a sequentially consistent full-map coherence protocol by as much as 41%. This is comparable to an implementation of weak consistency that uses a coalescing write-buffer to allow up to 16 outstanding requests for exclusive blocks. When used in conjunction with weak consistency DSI can exploit tear-off blocks-which eliminate both invalidation and acknowledgment messages-for a total reduction in messages of up to 26%.

DescriptorsCACHE-STORAGE; PROTOCOLS; SHARED-MEMORY-SYSTEMS; SOFTWARE-PERFORMANCE-EVALUATION.**Classification codes**C6150N Distributed-systems-software*;C5640 Protocols;C6120 File-organisation.**Keywords**

dynamic-self-invalidation; shared-memory-multiprocessors; **cache**- coherence-overhead-reduction; **invalidation-message-elimination**; local- **cache-block-copy**; conflicting-access; sequential-consistency; outstanding-copies; program-critical-path; hybrid-coherence-schemes; hardware; software; hardware-directory-based-write-invalidate-coherence-protocols; execution-time; sequentially-consistent-full-map-coherence-protocol; coalescing-write-buffer; outstanding-requests; exclusive-blocks; tear-off-blocks; **acknowledgment-message-elimination**.

Treatment codesP Practical.**Language**

English.

Publication typeConference-proceedings.**Availability**

CCCC: 0 89791 698 0/95/0006.\$3.50.

Digital object identifier

10.1109/ISCA.1995.524548.

Publication year

1995.

Publication date

19950000.

Edition

1995041.

Copyright statement

Copyright 1995 IEE.

COPYRIGHT BY The IET, Stevenage, UK

☒ **document 5 of 5** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0004013094 20051201.

Title

The Kyushu University reconfigurable parallel **processor-cache** architecture and **cache** coherence schemes.

Conference information

Proceedings of the International Symposium on Shared Memory Multiprocessing, Tokyo, Japan, 2-4 April 1991.

Sponsor(s): Inf. Process. Soc. Japan.

Source

Proceedings of the International Symposium on Shared Memory Multiprocessing, 1991, p. 218-29, 23 refs, pp. 251.

Publisher: Inf. Process. Soc. Japan, Tokyo, Japan.

Author(s)

Mori-S., Murakami-K., Iwata-E., Fukuda-A., Tomita-S.

Author affiliation

Mori, S., Murakami, K., Iwata, E., Fukuda, A., Tomita, S., Dept. of Inf. Syst., Kyushu Univ., Fukuoka.

Abstract

The Kyushu University reconfigurable parallel processor system is an MIMD-type multiprocessor which consists of 128 processing elements (PEs), interconnected by a full (128×128) crossbar network. The system employs reconfigurable memory architectures, a kind of local/remote memory architectures, and encompasses a shared-memory TCMP (tightly coupled multiprocessor) and a **message-passing LCMP** (loosely coupled multiprocessor). When the system is configured as a shared-memory TCMP, memory contentions will be obstacles to the performance. To relieve the effects, the system provides each PE with a private unified **cache**. Each PE may have the cached **copy** of shared data in its **cache** whether it accesses to local or remote memory, and therefore the multicache consistency, or **inter-cache** coherence, problem arises. The **cache** is a virtual-address direct-mapped **cache** to meet the requirements for the hit time and size. The virtual-address **cache** implementation causes the other consistency problem, the 'synonym' problem, called the **intra-cache** coherence problem. The paper presents the **cache** coherence schemes that have been chosen for resolving these **cache** coherence problems.

Descriptors

 **BUFFER-STORAGE;**  **MEMORY-ARCHITECTURE;**  **MULTIPROCESSING-SYSTEMS;**
 **PARALLEL-ARCHITECTURES;**  **STORAGE-MANAGEMENT.**

Classification codes

C5440 Multiprocessing-systems*;
C5220 Computer-architecture;
C5310 Storage-system-design;
C6120 File-organisation.

Keywords

Kyushu-University-reconfigurable-parallel-processor; **cache-** architecture; MIMD-type-multiprocessor; processing-elements; crossbar-network; reconfigurable-memory-architectures; shared-memory-TCMP; tightly-coupled-multiprocessor; **message-passing-LCMP**; loosely-coupled-multiprocessor; **inter-cache-coherence; intra-cache-coherence.**

Treatment codes

P Practical.

Language

English.

Publication type

Conference-proceedings.

Publication year

1991.

Publication date

19910000.

Edition

1991023.

Copyright statement

Copyright 1991 IEE.

COPYRIGHT BY The IET, Stevenage, UK



locally as:

PDF document



search strategy:

do not include the search strategy



Top - News & FAQs - Dialog

© 2006 Dialog